In the Claims

The following is a marked-up version of the claims with the language that is underlined ("___") being added and the language that contains strikethrough ("—") being deleted:

1. (Currently amended) A method for diagnosing faults in a system under test (SUT), the SUT defining data transmission paths through which data are transferred, said method comprising:

identifying ~~at least some~~ portions of the data transmission paths of the SUT capable of introducing errors in data transfer;

providing constraints defining relationships of at least some of the portions of the data transmission paths identified; and

diagnosing the SUT with respect to the constraints,

wherein identifying comprises providing a dataflow model corresponding to the SUT, the dataflow model including edges, each of the edges corresponding to a portion of one of the data transmission paths of the SUT capable of introducing errors in data transfer.

2. (Canceled).

3. (Currently amended) The method of claim [[2]] 1, wherein the dataflow model includes vertices, each of the edges being defined between two of the vertices.

4. (Original) The method of claim 3, wherein each of the vertices is at least one of a termination of an edge and representative of a location where an operation with respect to data can occur.

5. (Original) The method of claim 4, wherein the operation corresponding to a vertex includes at least one of dropping data, splitting data, routing data, replicating data and combining data.

6. (Original) The method of claim 4, further comprising:

receiving test results corresponding to the SUT; and

wherein diagnosing comprises analyzing the test results with respect to the dataflow model.

7. (Currently amended) The method of claim 6, wherein the SUT includes a counter[[s]] corresponding to at least [[some]] one of the edges of the dataflow model; and

further comprising:

receiving information, corresponding to the test results, from ~~at least some of~~ the counter[[s]].

8. (Original) The method of claim 6, wherein the dataflow model is a directed graph.

9. (Original) The method of claim 6, wherein analyzing the test results comprises:

receiving information corresponding to failed data transfers; and

identifying portions of the SUT potentially associated with the failed data transfers.

10. (Original) The method of claim 9, wherein analyzing the test results comprises:

exonerating portions of the SUT initially identified as being associated with the failed data transfers if those portions of the SUT are determined not to have initiated at least one of the failed data transfers.

11. (Original) The method of claim 1, wherein diagnosing the SUT comprises:

receiving information regarding data transfers with respect to the portions identified, the information being obtained via cyclic redundancy checking.

12. (Currently Amended) The method of claim 1, wherein ~~identifying comprises providing a dataflow model corresponding to the SUT,~~ the dataflow model ~~including edges and~~ includes vertices, ~~each of the edges corresponding to a portion of one of the data transmission paths of the SUT capable of introducing errors in data transfer,~~ each of the edges being defined between two of the vertices; and

wherein the constraints correspond to data flow characteristics of the SUT exhibited with respect to the vertices.

13. (Original) The method of claim 12, wherein at least one of the constraints of at least one of the vertices relates that an amount of data flowing into the vertex corresponds to an amount of data flowing from the vertex.

14. (Original) The method of claim 13, wherein the amount of data flowing into the vertex corresponds to an amount of at least one of: good data, bad data and a particular type of data flowing into the vertex.

15. (Currently amended) A method for diagnosing faults in a system under test (SUT), said method comprising:

providing a dataflow model representative of error-free behavior of the SUT, the dataflow model including information corresponding to a relationship of error detection capabilities of data packet flow through the SUT; [[and]]

providing constraints defining relationships of portions of the dataflow model, the constraints comprising equations describing the flow of the data packets through the SUT; and

diagnosing the SUT with respect to the dataflow model using the constraints.

16. (Canceled).

17. (Original) The method of claim 15, wherein diagnosing the SUT comprises:

generating information indicative of a manner of failure of the SUT.

18. (Currently amended) The method of claim 17, ~~wherein the flow of data is a flow of data packets; and~~

wherein diagnosing the SUT further comprises:

analyzing information acquired via cyclic redundancy checks performed at various locations associated with the flow of data.

19. (Canceled).

20. (Currently amended). A system for diagnosing faults in a system under test (SUT), said system comprising:

a dataflow model representative of error detection capabilities of the SUT; and

a reasoning engine associated with said dataflow model, said reasoning engine being adapted to evaluate test results corresponding to the SUT in relation to said dataflow model, wherein said dataflow model is a directed graph including edges and vertices, each of said edges corresponding to at least a portion of a data transmission path of the SUT through which an error can be introduced, each of said edges being defined by two of said vertices.

21. (Canceled).

22. (Currently amended). The system of claim [[19]] 20, wherein said reasoning engine is adapted to evaluate the test results of the SUT with respect to constraints, the constraints defining relationships of at least some of the portions of the dataflow model.

23. (Currently amended). The system of claim [[19]] 20, wherein said reasoning engine is adapted to receive information corresponding to failed data transfers and identify portions of the SUT potentially associated with the failed data transfers.

24. (Currently amended). The system of claim [[19]] 20, further comprising:

an SUT communicatively coupled to at least one of said dataflow model and said reasoning engine.

25. (Currently amended). A system for diagnosing faults in a system under test (SUT), said system comprising:

means for receiving test results corresponding ~~at least some~~ to portions of data transmission paths of the SUT; and

means for diagnosing the SUT with respect to constraints defining relationships of at least some of the portions of data transmission paths of the SUT, wherein said means for diagnosing includes means for analyzing the SUT with respect to a dataflow model representative of error-free behavior of the SUT.

26. (Canceled).

27. (Original) The system of claim 25, further comprising:

means for testing the SUT to generate test results.

28. (Currently amended). A diagnosis system stored on a computer-readable medium, the diagnosis system being adapted to diagnose faults in a system under test (SUT), said diagnosis system comprising:

logic configured to identify ~~at least some~~ portions of the data transmission paths of the SUT capable of introducing errors in data transfer;

logic configured to provide constraints defining relationships of at least some of the portions of the data transmission paths; and

logic configured to diagnose the SUT with respect to the constraints,

wherein said logic configured to diagnose comprises:

logic configured to provide a dataflow model representative of error-free behavior of the SUT; and

logic configured to analyze the SUT with respect to a dataflow model.

29. (Canceled).

30. (Original) The diagnosis system of claim 28, wherein said logic configured to diagnose includes logic configured to generate information indicative of the flow of data associated with a time of error detection.

31. (Original) The diagnosis system of claim 28, wherein said logic configured to diagnose includes logic configured to identify portions of the SUT potentially associated with failed data transfers.

32. (Original) The diagnosis system of claim 31, wherein said logic configured to diagnose includes logic configured to exonerate components initially identified as being associated with the failed data transfers.